VA Kidney Native iPhone App Build Screen Functionality Challenge - Deployment Guide

Revision History

Author	**Revision Number**	**Date**
TCCODER	1.0	Dec 25, 2017
TCCODER	1.1	Feb 04, 2018
TCCODER	1.2	Mar 04, 2018

[Deployment Instructions](#)

- [1. Deployment Dependencies](#)
- [2. Organization of Submission](#)
- [3. 3rd party Libraries](#)
- [4. Configuration](#)
  - [4.1. Configuration file](#)
  - [4.2. Sample data](#)
- [5. Deployment Instructions](#)
  - [5.2. Build and run the app in a simulator or on a real device](#)
- [6. Verification](#)
- [7. Resource Contact List](#)

Deployment Instructions

1. Deployment Dependencies

Before performing a deployment, it is assumed that the following have been set up:

- Xcode 9.2+
- OS X 10.12.6 or above
- iOS SDK 11 or above
- iPhone device or simulator with iOS 10+

2. Organization of Submission

- *src* – this directory contains the source code
- *src/VAKidneyNutrition.xcworkspace* – Xcode workspace to open.
- *docs* – this directory contains the documents for this application, including this deployment guide

3. 3rd party Libraries

SwiftyJSON - https://github.com/SwiftyJSON/SwiftyJSON
SwiftyJSON makes it easy to deal with JSON data in Swift. Version: 4.0.0

Charts - https://github.com/danielgindi/Charts
Version: 3.0.5

All libraries are configured in *src/Podfile*

4. Configuration

4.1. VAKidneyNutrition/Supporting Files/configuration.plist

You can access *configuration.plist* in Xcode in *VAKidneyNutrition.xcodeproj* project - *Supporing Files/configuration.plist*

configuration.plist file provides the following options:

- **ndbApiBaseUrl** - NDB base URL for API (see USDA nutrient database)
- **ndbApiKey** - NDB API key (see USDA nutrient database)
- **fdaApiBaseUrl** - FDA base URL for API (see FDA Drug Interaction and Product Labeling Database)
- **fdaApiKey** - FDA API key (see FDA Drug Interaction and Product Labeling Database)

4.2. Sample data

Sample data (used to fill the prototype with data) are stored in JSON files in *VAKidneyNutrition/Supporting Files/Sample Data/* group.
`allGoals.json` and `labValues.json` files added and define dependency of the generated goals and shown major lab values in Charts screen.

5. Deployment Instructions

5.2. Build and run the app in a simulator or on a real device

Pods directory should be pulled using the following command runned from `src` directory:

```
$ pod install
```

To build and run the app in a simulator or on a real device you will need to do the following:

1. Open *src/VAKidneyNutrition.xcworkspace* in Xcode
2. Select *VAKidneyNutrition* scheme from the top left drop down list.
3. Select a real iPhone (when connected) or a simulator from the top left dropdown list.
4. Click menu Product -> Run (Cmd+R)
5. Follow the verification steps in 7. Verification

6. Verification

Follow the challenge description and forum messages to verify the app. See some notes below. Also you can follow the video (how to launch the server and verify the screens) - https://youtu.be/rUsXsVh8r7Y.

Notes

- You must remove previously installed app on your device/simulator because Core Data model changed and requires fresh install.
- It's better to verify the app on a real device because HealthKit will contain some data.
- Recommendations and drug integration reports are generated after adding/editing meal based on current set of goals (of type "*.orderedSame*"). The goals has three types. Most of the goals related to Pills Intake have two thresholds (min, max) - type "*.orderedSame*". Other goals can be descending ("*.orderedDescending*"), e.g. "Distance", or ascending ("*.orderedAscending*"), e.g. "Weight Loss" (not presented in current version. The recommendations currently are generated for type "*.orderedSame*" only.
  The implementation of the logic that generate recommendations is described in classes in VAKidneyNutrition/Utils/Food group in Xcode project. Currently simple implementation is used (see `SimpleRecommendationSolver.swift`).
- Some profile data is saved to/retrieved from HealthKit: height, weight. Aga (birthday) is retrieved from HealthKit. After each update of the disease category and dialysis flag the set of goals is regenerated.
- Charts show data from HealthKit. However, if there are not much data the charts look poor. If there is only one value (the data are aggregated per months), then that single value is shown as round point. It's better if you enter some data for previous months for particular tested nutrition in advance in Health app.
- When meal is saved/updated the nutritions each food item is made from are saved into HealthKit. If meal is updated, then only the positive difference in nutritions are saved.

7. Resource Contact List

Name	Resource Email
TCCODER	Through TopCoder Member Contact